Explorando a Eficiência de Serviços de Conteinerização AWS*

Sara M. Cavalcante[1], **Cristina Boeres**[1], **Vinod E.F. Rebello**[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brazil

saracavalcante@id.uff.br, {boeres,vinod}@ic.uff.br

Resumo. *A alocação eficiente de recursos, bem como uma boa escolha de serviços da Nuvem para a execução de aplicações, é crucial para maximizar desempenho e minimizar custos. Este trabalho investiga o impacto do uso de diferentes serviços de conteinerização da AWS, explorando como a distribuição de recursos e carga de trabalho entre contêineres influencia o tempo e o custo da execução de uma aplicação de alinhamentos genéticos par-a-par.*

1. Introdução

A Computação em Nuvem vem se firmando como modelo de infraestruturas computacionais abrangente e sólido [Reed et al. 2023]. Empresas e instituições, outrora dependentes de servidores *On-premises*, migram para a nuvem devido à escalabilidade, à facilidade de gestão e ao pagamento sob demanda. A virtualização viabilizou essa tecnologia ao permitir a execução de várias máquinas virtuais (MVs) sobre um único servidor físico, provendo isolamento e portabilidade para diferentes aplicações e seus requisitos. Limitações e sobrecargas das MVs levaram ao surgimento dos contêineres, que, além de portáveis, compartilham o mesmo *kernel* do sistema operacional e oferecem maior eficiência de execução [Sharma et al. 2016]. Embora contêineres apresentem-se como a opção mais "leve" nesse cenário, a escolha do serviço e a alocação correta dos recursos computacionais são desafiantes e impactam os custos e o desempenho de aplicações.

Junto à consolidação da Computação em Nuvem, a realização de alinhamentos genéticos emerge como aplicação de alto desempenho que cientistas desejam executar sobre esse ambiente, em virtude da disponibilidade de recursos necessários e do pagamento somente pelo seu tempo de uso. Essa demanda ressalta a necessidade de análise dos serviços de conteinerização em nuvem aplicados para o estudo da similaridade genética, o qual é fundamental no desenvolvimento de estratégias de combate a surtos de doenças e na confecção de novas vacinas, por exemplo. Nesse contexto, o *Multi-Platform Architecture for Sequence Aligner* (MASA), capaz de realizar alinhamentos de sequências com mais de 200 milhões de nucleotídeos em plataformas distintas [De O. Sandes et al. 2016], destaca-se. Desse modo, a versão MASAOpenMP foi aqui adotada para a análise dos serviços da AWS (*Amazon Web Services*): ECS (*Amazon Elastic Container Service*) junto ao EC2 (*Amazon Elastic Compute Cloud*) e ao AWS Fargate, de forma disjunta.

2. Objetivos do Estudo e Configurações dos Serviços da AWS Utilizados

Esses serviços foram avaliados na região us-east-1 da AWS, na aplicação MASAOpenMP em configurações idênticas de contêineres de mesma imagem *Docker* sobre ambiente Linux-ARM64, apresentadas na Tabela 1. Ressalta-se que o somatório de vCPUs em

*Financiado pelo projeto CNPq/AWS (processo 421828/2022-6).

cada configuração da Tabela 1 totaliza 10, pois essa é a quantidade máxima de vCPUs que pode ser atribuída à tarefa no caso de uso de ECS junto ao EC2[1]. Em termos de recursos necessários, utilizou-se, em ECS-EC2 Sob demanda e ECS-EC2 Spot, a instância c7g.4xlarge, considerando a opção de alocação *Preço mais baixo* em caso de *Spot*. Já em ECS-Fargate, por ser necessário definir um "tamanho de tarefa" em função de recursos de CPU e memória, foi escolhida a opção predefinida de 16 vCPUs e 32 GB, objetivando uma capacidade de computação proporcional à da instância c7g.4xlarge.

Sequências da base de dados *Genbank* do NCBI (*National Center of Biotechnology Information*) foram coletadas a fim de formar grupos $g \in G$ de oito sequências de tamanho g, onde $G = \{200k, 300k, 400k\}$. Executou-se cada tarefa com $c \in C$ contêineres, onde $C = \{1, 2, 4\}$. Assim, denota-se por uma configuração a tupla (g, c), onde $g \in G$ e $c \in C$. Cada configuração em $G \times C$ foi executada três vezes, de modo que o tempo de execução e o custo estimado são calculados em função da média dos tempos.

Cada tarefa de configuração (g, c) realizou alinhamentos par-a-par entre as sequências de um grupo g: $\forall i = 1, 2, \ldots, 8$, a i-ésima sequência de g foi alinhada com cada j-ésima sequência de g, onde $i < j$, totalizando 28 alinhamentos. Essa carga foi dividida entre os c contêineres, de forma que cada contêiner executou $\frac{28}{c}$ alinhamentos em sequência, utilizando tantos *threads* quanto seu número de vCPUs para cada alinhamento.

Tabela 1. Distribuição de recursos e cargas de trabalho entre contêineres

nº de contêineres	ID do contêiner	vCPUs	GB	nº de alinhamentos
1	0	10	20	28
2	0	5	10	14
	1	5	10	14
4	0	3	6	7
	1	3	6	7
	2	2	4	7
	3	2	4	7

O custo médio de ECS-EC2-Sob Demanda e ECS-Fargate foi estimado com base nos preços de 18/09/2024. Em ECS-EC2-Sob Demanda, foi utilizado o preço de c7g.4xlarge, 0.58 USD/h, enquanto em ECS-Fargate foram aplicados os preços por vCPU, 0.03238 USD/h, e por GB, 0.00356 USD/h, conforme o tamanho da tarefa. Assim, nesses casos, o custo de cada configuração (g, c) é um produto entre o tempo médio demandado por (g, c) e os preços coletados. Por outro lado, em razão da flutuação dos preços de ECS-EC2-Spot, o custo para cada configuração (g, c) é dado pela média entre os custos de cada tarefa (g, c), sendo cada um deles o produto entre o tempo demandado e o preço mais baixo de c7g.4xlarge *Spot* vigente no momento do início da tarefa.

3. Resultados e Análise Experimental

Resultados relativos a cada configuração (g, c) se encontram na Tabela 2, na qual a diminuição percentual de tempo $d(g, c)$ é dada em relação ao tempo médio da configuração $(g, 4)$. A Tabela 2 indica que ECS-EC2-Spot oferece os menores custos entre os serviços, com tempos de execução semelhantes ao ECS-EC2-Sob Demanda. Em contrapartida, salienta-se que, quanto maior o tempo de execução da tarefa, maior a probabilidade de revogação de instâncias *Spot*, o que exige que o usuário prepare sua

[1] `docs.aws.amazon.com/pt_br/AmazonECS/latest/developerguide/`

aplicação para lidar com tal evento. Nota-se que ECS-Fargate apresenta o pior desempenho de tempo dentre os serviços, dado que o EC2 é 13,01% mais rápido, com base na média de tempo entre ECS-EC2-Sob Demanda e ECS-EC2-Spot.

Tabela 2. Tempo (em min:seg), diminuição percentual de tempo ($d(g, c)$) e custo (em USD) estimado entre EC2-Sob Demanda, EC2-Spot e Fargate com ECS.

Config.		ECS-EC2-Sob Demanda			ECS-EC2-Spot			ECS-Fargate		
g	c	Tempo	$d(g,c)$	Custo	Tempo	$d(g,c)$	Custo	Tempo	$d(g,c)$	Custo
200k	1	04:56.43	18.03	0.04776	04:56.87	16.85	0.01197	06:36.19	-	0.06955
	2	04:56.47	18.02	0.04776	04:52.80	17.99	0.01180	05:58.75	3.65	0.06298
	4	06:01.65	0.00	0.05827	05:57.04	0.00	0.01438	06:12.33	0.00	0.06536
300k	1	10:42.28	18.66	0.10348	10:42.85	19.53	0.02590	12:35.99	6.07	0.13272
	2	10:50.52	17.62	0.10481	10:44.23	19.36	0.02595	12:34.54	6.25	0.13246
	4	13:09.67	0.00	0.12722	13:18.88	0.00	0.03218	13:24.87	0.00	0.14130
400k	1	18:51.55	18.79	0.18231	18:52.80	18.61	0.04564	21:57.83	19.00	0.23135
	2	18:55.00	18.55	0.18286	18:55.35	18.42	0.04574	21:48.21	19.59	0.22966
	4	23:13.41	0.00	0.22449	23:11.74	0.00	0.05607	27:06.92	0.00	0.28561

Para MASAOpenMP, no entanto, as análises apontam que a grande vantagem de executar no ECS-Fargate está no fato de que o usuário não precisa escolher dentre a variedade de instâncias virtuais disponíveis no EC2, embora seja necessário conhecer o número de vCPUs e a quantidade de memória demandados pela sua aplicação nos dois casos. Por outro lado, ao comparar com a execução de ECS-EC2-Spot, o custo monetário associado à execução de ECS-Fargate mostrou-se maior: de 4,6 a 5,8 vezes no caso de sequências de 200k; de 4,4 a 5,1 vezes no caso de 300k; e 5,1 vezes no caso de 400k. Já o custo de ECS-EC2-Sob demanda foi aproximadamente 4 vezes superior em comparação ao ECS-EC2-Spot para todos os tamanhos de sequências.

4. Conclusões e Análises Futuras

A análise comparativa mostrou que o EC2 *Spot* oferece o melhor custo-benefício para cargas de curto prazo, enquanto *Sob Demanda* garante maior estabilidade a um custo mais elevado. Todavia, a limitação de 10 vCPUs por tarefa no EC2 pode tornar inviável a utilização do ECS para aplicações que exigem contêineres com mais vCPUs. Por sua vez, AWS Fargate apresenta desempenho inferior ao passo que revela-se ser a opção mais simples em usabilidade. Elenca-se, para análises futuras, o estudo do comportamento dos serviços analisados neste trabalho sobre *clusters* a fim de verificar sua eficiência em processamento distribuído e um comparativo com EKS (*Amazon Elastic Kubernetes Service*).

Referências

De O. Sandes, E. F., Miranda, G., Martorell, X., Ayguade, E., Teodoro, G., and De Melo, A. C. M. A. (2016). MASA: A multiplatform architecture for sequence aligners with block pruning. *ACM Transactions on Parallel Computing*, 2(4).

Reed, D., Gannon, D., and Dongarra, J. (2023). HPC Forecast: Cloudy and uncertain. *Commun. ACM*, 66(2):82–90.

Sharma, P., Chaufournier, L., Shenoy, P., and Tay, Y. C. (2016). Containers and virtual machines at scale: A comparative study. In *Proceedings of the 17th International Middleware Conference*, Middleware '16, New York, NY, USA. Association for Computing Machinery.